

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-229217

(P2001-229217A)

(43) 公開日 平成13年8月24日 (2001.8.24)

(51) Int. Cl.

G 0 6 F 17/50

識別記号

F I

G 0 6 F 15/60

テーマコード (参考)

6 5 4 M 5 B 0 4 6

6 5 6 B

審査請求 未請求 請求項の数7 O L (全 19 頁)

(21) 出願番号 特願2000-42083 (P2000-42083)

(22) 出願日 平成12年2月18日 (2000.2.18)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 岡田 和久

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72) 発明者 西田 浩一

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(74) 代理人 100078282

弁理士 山本 秀策

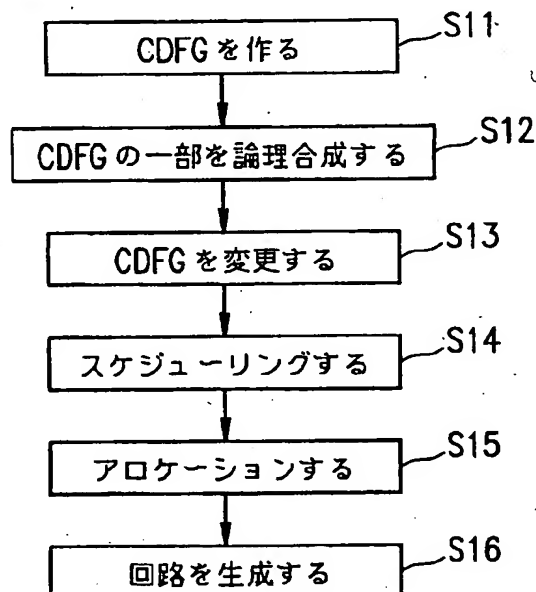
Fターム (参考) 5B046 AA08 BA03

(54) 【発明の名称】 高位合成方法およびその実施に使用される記録媒体

(57) 【要約】

【課題】 高位合成に必要な処理時間を短くし、高位合成時において、遅延時間の見積り精度を高めることができ、高位合成した回路の動作速度を速くすることができ、さらに、高位合成した回路の規模を小さくすることができる。

【解決手段】 処理の動作のみを記述した動作記述を、演算を示す節点とデータの流れを示す入出力枝とによって構成されるコントロールデータフローグラフ (C D F G) に変換する際に、C D F Gに含まれる部分C D F Gを、予め論理合成し、論理合成して得られる回路を1つの節点として扱う。



【特許請求の範囲】

【請求項1】 処理の動作のみを記述した動作記述を、演算を示す節点とデータの流れを示す入出力枝とによって構成されるコントロールデータフローグラフ(CDFG)に変換するCDFG変換工程と、
該CDFG変換工程にて得られたCDFGをスケジューリング工程と、
該スケジューリング工程にてスケジューリングされたCDFGの実行に必要な演算器、セレクタ等をそれぞれ割り当てるアロケーション工程とを包含する高位合成方法であって、
前記CDFG変換工程において、CDFGに含まれる部分CDFGを、予め論理合成し、論理合成して得られる回路を1つの節点として扱うことを特徴とする高位合成方法。

【請求項2】 前記部分CDFGは、条件分岐に対応したものである請求項1に記載の高位合成方法。

【請求項3】 前記部分CDFGは、条件が成立する場合としない場合の処理が同一のクロック周期で終了するものである請求項2に記載の高位合成方法。

【請求項4】 前記部分CDFGに含まれる特定の節点を、条件分岐に対応するCDFGの外部に配置する請求項2に記載の高位合成方法。

【請求項5】 前記部分CDFGを分割することによって、その部分CDFGに含まれる特定の節点を、該部分CDFGの外部に配置する請求項2記載の高位合成方法。

【請求項6】 前記アロケーション工程において前記部分CDFGに同一の演算器が複数割り当てられる場合に、その1つの演算器の面積がセレクタの面積の定数倍以上の場合に、その演算器に対応する節点を、条件分岐に対応するCDFGの外部に配置する請求項4または請求項5に記載の高位合成方法。

【請求項7】 請求項1に記載の高位合成方法の手順がプログラムされていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はLSIの動作記述からデジタル回路を自動合成する高位合成方法およびその実施に使用される記録媒体に関する。

【0002】

【従来の技術】 従来から、ASIC(Application Specific Integrated Circuits: 特定用途向けIC)の設計等、短期間の設計を要求される場合に、高位合成方法が特に有効な技術として知られている。

【0003】 高位合成とは、ハードウェアの構造に関する情報を含まずに、処理のアルゴリズムのみを記述した動作記述から、回路を自動的に合成する技術である。なお、高位合成技術の詳細を記した文献としては、「K1

uwer Academic Publishers」による「High Level Synthesis」等がある。

【0004】 以下、例えば、次の(1)式に示す動作記述から回路を自動的に高位合成する方法について説明する。

$$【0005】 x = (a + b) * (b + c) \dots (1)$$

高位合成方法は、通常、図1に示すフローチャートの手順に従って実施され、まず、動作記述の実行の制御の流れと、データのの流れとを解析し、CDFG(Control Data Flow Graph)と呼ばれるモデルに変換する(図1のステップS1参照)。CDFGとは、プログラムのフローチャートに類似したグラフである。CDFGは、節点と入出力枝とによって構成される。入出力枝は、データまたはコントロール信号を表し、節点は演算を表わす。演算の入出力は、節点の入出力枝に対応する。

【0006】 例えば、(1)式に示す動作記述は、図2に示すCDFGで表わされる。図2のCDFGは、加算を表わす2つの第1および第2の加算節点11および12と、乗算を表わす1つの乗算節点13を含み、入力「a」および「b」を加算した結果と、入力「b」および「c」を加算した結果とを乗算し、その結果が出力「x」であることを表している。

【0007】 (1)式に示す動作記述が図2に示すCDFGに変換されると、スケジューリングが実施される(図1のステップS2参照)。スケジューリングとは、CDFGの節点に対応する演算をいつ実行するか、すなわち、どのクロックステップで実行するかを決定することである。この場合に、各演算の遅延時間を考慮して、全ての節点がクロック周期内に収まるようにする必要がある。

【0008】 図2のCDFGをスケジューリングした一例を図3に示す。図3のスケジューリング結果では、2つの加算と1つの乗算を、1つのクロックステップ(step1)で実行するようスケジューリングされている。この場合、データ依存関係にある演算の遅延時間の合計は、1つのクロックステップのステップ周期を超えないようにスケジューリングされる。例えば、加算器の遅延時間が5(nsec)、乗算器の遅延時間が60(nsec)の場合、クロック周期が65(nsec)以上の場合には、図3に示すように、1つのクロックステップ(step1)に全ての演算をスケジューリングすることができる。

【0009】 また、異なるクロックステップにそれぞれスケジューリングされた同一の演算は、1つの演算器によって実行することができる。このために、図5に示すように、第1の加算を示す第1の加算節点11と、第2の加算を示す第2の加算節点12とを、異なるクロックステップstep1およびstep2にてスケジューリ

ングして、1つの加算器を生成することもできる。この場合、前述したように、クロック周期が65 (nsec) 以上であれば、遅延時間が5 (nsec) の第2の加算と、遅延時間が60 (nsec) の乗算とを、第2のクロックステップstep 2にスケジューリングしても問題はない。

【0010】動作記述がスケジューリングされると、アロケーションが実行される(図1のステップS3参照)。アロケーションとは、スケジューリングされたC D F Gの実行に必要な演算器およびレジスタをそれぞれ生成して、C D F Gの演算に演算器を割り当て、また、隣接するクロックステップの境界を横切る入出力枝に、レジスタ、セクタ等を割り当てる処理である。そして、アロケーション処理することによって、(1)式に示す動作記述を実行する回路が生成される(図1のステップS4参照)。

【0011】図3のスケジューリング結果は、図4に示すように、第1および第2の加算に対して、第1の加算器14および15がそれぞれ生成されて、第1および第2の加算節点11および12に割り当てられるとともに、乗算に対して乗算器16が生成されて、乗算節点13に割り当てられる。この場合、隣接するクロックステップの境界を横切る入出力枝が存在しないために、レジスタおよびセクタは生成されない。

【0012】これに対して、図5のスケジューリング結果では、図6に示すように、異なるクロックステップstep 1および2に、第1および第2の加算に対する第1および第2の加算節点11および12がそれぞれスケジューリングされているために、これらの加算節点11および12に対して、1つの加算器14のみが生成される。また、クロックステップstep 1およびstep 2の境界を横切る入出力枝21および22が存在するために、各入出力枝21および22に対して、セクタ(図において「sel」で示す)23およびレジスタ(図においては、「reg」で示す)24がそれぞれ生成される。

【0013】図6に示す回路では、入力「a」および「b」がセクタ23にそれぞれ入力されて、セクタ23の出力が加算器14に与えられている。加算器14には、入力「b」も与えられており、加算器14の出力が、レジスタ24および乗算器16にそれぞれ与えられている。この場合、生成されたセクタ23およびレジスタ24それぞれに対するコントロール信号を生成するコントローラ(図において「controller」で示す)25が生成される。コントローラ25は、セクタ23に対してセクタ信号k1を出力し、セクタ23は、セレクト信号k1が「1」の場合に入力「a」を、セレクト信号k1が「0」の場合に入力「b」をそれぞれ出力する。また、コントローラ25は、レジスタ24に対してイネーブル信号k2を出力し、レジスタ2

4は、イネーブル信号k2が「1」の場合に、クロックが立ち上がる瞬間の入力の値を記憶し、イネーブル信号k2が「0」の場合には、記憶している値を保持して、その保持された値を出力する。

【0014】図6の回路の動作を説明する。クロックステップstep 1では、セレクト信号k1とイネーブル信号k2は、共に「1」とされ、セクタ23は、入力「a」を選択して加算器14に出力する。これにより、加算器14は、各入力「a」および「b」の加算値「a+b」をレジスタ24に出力する。レジスタ24は、イネーブル信号k2が「1」であるために、レジスタ24の出力「a+b」が記憶される。

【0015】クロックステップstep 2では、セレクト信号k1およびイネーブル信号k2は、共に「0」とされ、セクタ23は、入力「c」を選択して加算器14に出力する。これにより、加算器14は、各入力「c」および「b」の加算値「c+b」を乗算器24に出力する。この場合、レジスタ24に対するイネーブル信号k2が「0」になっているために、保持されている値「a+b」を乗算器16に出力する。乗算器16は、与えられる値「c+b」および「a+b」を乗算して、その積をxとして出力する。

【0016】図4に示す回路では、1クロックステップ(たとえば100nsec内)によって動作記述の処理が終了するが、2つの加算器14および15が2つ必要である。これに対して、図6に示す回路では、2クロックステップ(たとえば200nsec)が必要であるが、1つの加算器14を使用するだけでよい。このように、高位合成方法では、高速な回路が必要な場合には、図4に示す回路、小面積な回路が必要な場合には、図6に示す回路がそれぞれ生成される。

【0017】図5に示すスケジューリング結果では、2つの加算を1つの加算器で実行できることが理解される。しかしながら、演算が多くなると、どの演算とどの演算を1つの演算器で実行するかを決定する手順が必要になる。この手順がアロケーションである。スケジューリング、アロケーションは、高位合成の分野では周知である(前述の「High Level Synthesis」参照)。

【0018】次に、動作記述が条件分岐を含む場合について説明する。条件分岐を含む動作記述を高位合成する方法は、条件分岐を制御するためのコントロール信号を生成する必要があるが、基本的には、条件分岐を含まない図1の高位合成方法と同様である。この方法としては、例えば、特開平11-250112号公報に開示されている。

【0019】以下、図7に示す条件分岐を含む動作記述の高位合成方法について説明する。図7に示す動作記述では、条件(if)「d>e」の場合は、 $x=a-b$ であり、その他の条件の場合(else)は、 $x=a*$

(b-c)である。この場合のCDFGをスケジューリングすると、図8に示すように、条件分岐としてIF節
点32を含むものとなる。

【0020】図8に示すように、クロックステップstep 1には、比較を示す比較節点31がスケジューリングされており、クロックステップstep 2には、IF節
点32がスケジューリングされている。比較節点31には、入力「d」および「e」からの入出力枝が接続されており、また、コントロール信号kを与えるコントロール
入出力枝33が接続されている。図8では、コントロール入出力枝33を点線で表している。

【0021】また、クロックステップstep 2にスケジューリングされたIF節点32は、内部に第1および第2の副CDFG34および35を有している。第1の副CDFG34は、IF節点を含むことができる。第1の副CDFG34は、コントロール信号kが「1」

(真)の時に実行され(「true」で表す)、第2の副CDFG35は、コントロール信号kが「0」(偽)の時に実行される(「false」で表す)。このように、比較節点31およびIF節点32は、コントローラが必要になるために、それぞれ異なるクロックステップにスケジューリングされている。

【0022】「true」で表された副CDFG34は、入力「a」および「b」が入力される第1の減算節点36が設けられており、「false」で表される副CDFG35には、第2の減算節点37および乗算節点38がそれぞれ設けられている。第2の減算節点37には、入力「b」および「c」からの入出力枝が接続されており、乗算節点38には、入力「a」および第1の減算節点37からの入出力枝が接続されている。各副CDFG34および35は、同時に実行されないため、それぞれに設けられた第1および第2の各減算節点36および37に対して1つの減算器45(図9参照)がアロケーションされる。

【0023】図8に示すスケジューリング結果を実行する回路は、図9のようになる。図9に示す回路では、比較節点31として、比較器41がアロケーションされており、入力「a」および「b」に対しては第1のセクタ43が、また、入力「b」および「c」に対しては、第2のセクタ44が割り当てられている。また、第1および第2の減算節点36および37に対して、1つの減算器45が割り当てられており、乗算節点38に対して乗算器46が割り当てられている。そして、減算器45および乗算器46の出力に対して第3のセクタ47が割り当てられている。第1~第3の各セクタ43、44、47は、コントローラ42からのコントロール信号kによって、それぞれ制御される。コントロール信号kが値「1」の場合には、第1のセクタ43は、入力「a」、第2のセクタ44は、入力「b」、第3のセクタ47は、減算器45の出力を、それぞれ選択す

る。

【0024】このような回路の動作について説明する。クロックステップstep 1において、入力「d」および「e」が、比較器41により比較され、その結果が、コントローラ42に入力される。コントローラ42は、比較器41から入力される比較結果に基づいて、コントロール信号kを生成し、次のクロックステップstep 2において出力する。コントロール信号kは、比較器41の比較結果が、「 $d < e$ 」の場合には、値「1」

(真)を出力し、それ以外の場合には、値「0」(偽)を出力する。クロックステップstep 2において、コントロール信号kが値「1」の場合には、第1および第2のセクタ41および42は、それぞれ、入力「a」および「b」を選択するために、減算器45は、「 $a - b$ 」の減算を実行し、第3のセクタ47は、減算器45の出力を選択して、減算器45による減算結果「 $a - b$ 」が出力「x」とされる。

【0025】これに対して、コントロール信号kの値が「0」の場合には、第1および第2のセクタ43および44は、入力「b」および「c」をそれぞれ選択して、減算器45は、「 $b - c$ 」の減算を実行する。そして、この減算結果が、乗算器46に与えられて、乗算器は、減算器45による減算結果と、入力「a」との乗算結果を第3のセクタ47に出力する。第3のセクタ47は、乗算器46の出力を選択して、乗算器46による乗算「 $a * (b - c)$ 」が出力「x」とされる。

【0026】

【発明が解決しようとする課題】高位合成方法におけるスケジューリングおよびアロケーションは、非常に複雑な処理である。このため、1つの動作記述が多数の演算を含む場合、CDFG中の節点数が多数になり、計算機を用いて高位合成を行う場合に、非常に長い処理時間が必要になる。

【0027】また、高位合成に際しては、CDFG中の各節点の遅延時間に対応する演算器の遅延時間に基づいて、演算時間が算出される。例えば、図10に示すように、クロック周期の時間が、例えば10(nsec)内に、遅延時間が2(nsec)の第1および第2の「NOT」節点51および52と、遅延時間が8(nsec)の「AND」節点53がスケジューリングされている場合に、このスケジューリング結果に対応する回路は、図11(a)のように、「NOT」節点51および52に対して「NOT」演算器54および55がそれぞれ割り当てられるとともに、「AND」節点53に、「AND」演算器56が割り当てられる。

【0028】この場合、論理合成によって、図11(b)に示すように、一対の「NOT」演算器51および52と、1つの「AND」演算器とが、1つの「NOR」演算器57に最適化される可能性がある。その結果、1つの「NOR」演算器57の遅延時間は6(nsec)

sec) になり、高位合成時において見積もられた遅延時間10 (nsec) とは誤差が生じることになる。すなわち、高位合成時においては、遅延時間を必要以上に大きく見積ることになり、CDFG全体をスケジューリングした際に、必要なステップ数が必要以上に増加することになり、生成される回路の動作が遅くなる可能性がある。

【0029】さらに、高位合成では、図6に示すように、演算器である加算器14の前にセクタ23が挿入される可能性があるが、このように、セクタ23が加算器14等の演算器の前に挿入されることにより、遅延時間が増加して、回路が正常に動作しなくなるおそれがある。

【0030】また、図7に示すように、条件分岐を含む動作記述を高位合成する場合に、条件分岐節点の入出力枝が多数になっていると、図9に示すように、合成した回路が多数のセクタを含むことになり、回路規模が大きくなるという問題もある。

【0031】さらにまた、図9に示すように、条件分岐のコントロール信号をコントローラ42で生成する場合は、コントローラ42は、通常、クロックに同期して動作するため、条件分岐本体を実行する前のクロックステップにおいて、条件を決定しておく必要がある。従って、条件分岐のコントロール信号を生成するクロックステップと、条件分岐本体を実行するクロックステップとを、図8に示すように、別にすることが必要である。その結果、回路動作を高速化することができなくなるおそれもある。

【0032】本発明は、このような問題を解決するものであり、その目的は、高位合成に必要な処理時間を短くすることができるとともに、高位合成時において、遅延時間の見積り精度を高めることができ、また、高位合成した回路の動作速度を速くすることができるとともに、高位合成した回路の規模を小さくすることができる高位合成方法を提供することにある。

【0033】

【課題を解決するための手段】上記の問題を解決するために、本発明の請求項1記載の高位合成方法は、処理の動作のみを記述した動作記述を、演算を示す節点とデータの流れを示す入出力枝とによって構成されるコントロールデータフローグラフ(CDFG)に変換するCDFG変換工程と、該CDFG変換工程にて得られたCDFGをスケジューリング工程と、該スケジューリング工程にてスケジューリングされたCDFGの実行に必要な演算器、セクタ等をそれぞれ割り当てるアロケーション工程とを包含する高位合成方法であって、前記CDFG変換工程において、CDFGに含まれる部分CDFGを、予め論理合成し、論理合成して得られる回路を1つの節点として扱うことを特徴とする。

【0034】本発明の請求項2記載の高位合成方法で

は、前記部分CDFGは、条件分岐に対応したものである。

【0035】本発明の請求項3記載の高位合成方法では、前記部分CDFGは、条件が成立する場合としない場合の処理が同一のクロック周期で終了するものである。

【0036】本発明の請求項4記載の高位合成方法では、前記部分CDFGに含まれる特定の節点を、条件分岐に対応するCDFGの外部に配置する。

【0037】本発明の請求項5記載の高位合成方法では、前記部分CDFGを分割することによって、その部分CDFGに含まれる特定の節点を、該部分CDFGの外部に配置する。

【0038】本発明の請求項6記載の高位合成方法では、前記アロケーション工程において前記部分CDFGに同一の演算器が複数割り当てられる場合に、その1つの演算器の面積がセクタの面積の定数倍以上の場合に、その演算器に対応する節点を、条件分岐に対応するCDFGの外部に配置する。

【0039】本発明の請求項7記載の記録媒体は、請求項1に記載の高位合成方法の手順がプログラムされていることを特徴とする。

【0040】

【発明の実施の形態】本発明の実施の一形態について図面に基いて説明する。

【0041】<実施の形態1>本実施の形態の高位合成方法は、図12に示すフローチャートの手順に従って実施される。以下、図7に示す動作記述に基づいて高位合成を行なう場合について説明する。図7に示す動作記述は、条件(if)「 $d > e$ 」の場合は、 $x = a - b$ であり、その他の条件の場合(else)は、 $x = a * (b - c)$ である。

【0042】本実施の形態の高位合成方法では、まず、このような動作記述の制御の流れと、データの流れとを解析して、動作記述を、CDFG(Control Data Flow Graph)と呼ばれるモデルに変換する(図12のステップS11参照、以下同様)。そして、CDFGの一部を論理合成することによって(ステップS12)、CDFGを変更する(ステップS13)。その後は、従来の高位合成方法と同様に、変更されたCDFGをスケジューリングし(ステップS14)、その後にアロケーションすることによって(ステップS15)、回路を生成する(ステップS16)。

【0043】図7に示す動作記述のCDFGをスケジューリングすると、図8に示すように、条件分岐としてIF節点32を含むものとなる。図8に示すスケジューリング結果は、従来技術において説明した通りであるので、その説明を省略する。

【0044】本実施の形態では、図8に示すスケジューリング結果において、CDFGの一部であるIF節点3

2内を論理合成して組合せ論理回路を生成する。図13は、1F節点32において、論理合成して生成された組合せ論理回路を示している。

【0045】図13に示す第1の部分回路61は、1F節点32の条件が「真」の場合に実行される副CDFG34（図8参照）を、組合せ論理回路で実現したものである。この第1の部分回路61には、第1の減算器63が設けられており、第1の減算器63に、入力「a」および「b」がそれぞれ入力されている。

【0046】また、図13に示す第2の部分回路62は、1F節点32の条件が「偽」の場合に実行される副CDFG35（図8参照）を、組合せ論理回路で実現したものである。この第2の部分回路62には、第2の減算器64および乗算器65がそれぞれ設けられており、第2の減算器64に、入力「b」および「c」が入力されている。乗算器65には、入力「a」と第2減算器64の出力とが、入力されている。

【0047】また、第1の減算器63の出力は、第1の部分回路61の出力になっており、乗算器65の出力は、第2の部分回路62の出力になっている。そして、第1および第2の部分回路61および62の出力が、セクタ66に入力されており、セクタ66の出力が回路全体の出力とされている。セクタ66のセレクト信号kは、1F節点32のコントロール入力、コントローラを介することなく直接セレクト66に入力されている。

【0048】次に、1F節点32内を論理合成して得られる組合せ論理回路に対応する節点を準備する。図13に示す1F節点32は、図14に示すように、CIF（Combinational IF）節点67とされる。このCIF節点67は、1F節点32の組合せ論理回路における4つの入力「k」、「a」、「b」、「c」と1つの出力「x」に対応して、4つの入力枝と1つの出力枝が設けられる。

【0049】図13に示す回路は、組合せ論理回路であるため、CDFGにおいては、加算や減算などの演算と同様に扱うことができる。

【0050】図8に示すCDFGの1F節点32を、図14に示すCIF節点67に置き換えると、図15に示すCDFGとなる。図8のCDFGは、比較節点31、2つの減算節点36および37、乗算節点38、および、1F節点31の合計5つの節点を含むのに対して、1F節点32をCIF節点67に置き換えることによって、図15に示すCDFGでは、比較節点31とCIF節点67の2つの節点しか含まない。このように、本実施の形態によりCDFGに含まれる節点数が減少し、高位合成時のスケジューリング、アロケーションなどの処理に必要な時間を短くすることができる。

【0051】また、図15に示すCIF節点67に対応する回路は既に論理合成されているため、図13に示す

回路における各入力「a」、「b」、「c」、「k」から出力「x」への遅延時間として、セクタ66の遅延時間も含めて、論理最適化後の最終的な値が求まる。よって、図14に示すCIF節点67の遅延時間は正確なものであり、図15において、1クロックステップにスケジューリングされた節点の遅延時間の合計が、クロック周期よりも大きくなって回路が誤作動するおそれがなく、反対に、クロック周期よりも小さくなり過ぎることによって、回路の動作に無駄が生じて、回路の動作が遅くなるおそれもない。

【0052】さらに、図8に示すCDFGの1F節点32には、コントロール入力枝33が設けられているために、1F節点32のコントロール信号kを生成する比較節点31と1F節点32とは別のクロックステップにスケジューリングする必要がある。これに対して、図14に示すCIF節点67では、コントロール信号kがコントロールを介さない通常の入力枝33によって与えられるため、加算や乗算などの通常の演算の節点と同じようにスケジューリングすることができる。このため、図15に示すように、コントロール信号kを生成する比較節点31とCIF節点67とを同一のクロックステップにスケジューリングすることができる。これにより、生成される回路の動作が高速化される。

【0053】図15に示すスケジューリング結果から、図16に示す回路が得られる。図16において、点線で示す回路68は、図14に示すCIF節点67に対応する組合せ論理回路（図13の1F節点32に対応）である。

【0054】本実施の形態の高位合成方法では、CIF節点67に対応する回路を予め論理合成するため、CIF節点67内で使われる演算器を他の演算で用いられる演算器と共通化することはできない。しかし、本実施の形態の高位合成方法では、CIF節点67に対応する組合せ論理回路がセクタを必要としないため、演算器を共通化できないことによって回路規模が大きくなることよりも、セクタを減少させることによって回路規模が小さくなることの方が有利になる場合には、好適である。

【0055】例えば、従来の高位合成方法によって得られる図9に示す回路構成では、1つの減算器45のみが設けられているが、3つのセクタ43、44、47を必要とする。これに対して、図16に示す回路構成では、2つの減算器63および64が必要であるが、1つのセクタ66のみでよいために、2つセクタの回路規模が、1つの減算器の回路規模よりも小さい場合は、本実施の形態の高位合成方法により、図16に示す回路構成とすることによって、図9に示す回路構成よりも、全体の回路規模を小さくすることができる。

【0056】また、図16に示す回路構成では、図9に示すコントローラ42が不要となるために、これによっても、全体の回路規模を小さくすることができる。

【0057】＜実施の形態2＞次に、本発明の高位合成方法の実施の形態の他の例を、図17の動作記述に基づいて説明する。図17に示す動作記述は、図7に示す動作記述、すなわち、条件 (if) 「 $d > e$ 」の場合、 $x = a - b$ 、その他の条件の場合 (else)、 $x = a * (b - c)$ という動作記述に、条件 (if) 「 $f > g$ 」の場合、 $n = y - x$ 、その他の条件の場合 (else)、 $n = x * y * z$ という動作記述が加えられている。

【0058】図17に示す動作記述は、CDFGに変換されて、図18に示すようにスケジューリングされる。クロックステップstep1およびstep2には、図17に示す動作記述において、図7に示す動作記述と同様の部分が、図8と同様にスケジューリングされており、その説明については省略する。

【0059】クロックステップstep2には、比較を示す比較節点71もスケジューリングされており、クロックステップstep3には、IF節点72がスケジューリングされている。比較を示す比較節点71には、入力「f」および「g」からの入出力枝がそれぞれ接続されている。比較節点71からの入出力枝は、コントロール信号k2を与える制御入出力枝として、第2のIF節点72に接続されている。

【0060】第2のIF節点72の内部には、第1および第2の副CDFG74および75が設けられている。第1の副CDFG74は、コントロール信号k2が「1」（真）の場合に実行され（「true」で表す）、第2の副CDFG75は、コントロール信号k2が「0」（偽）の場合に実行される（「false」で表す）。

【0061】「true」で表された副CDFG74は、入力「y」および「z」からのそれぞれの入出力枝が接続された減算節点76が設けられており、「false」で表される副CDFG75には、第1の乗算節点77および第2の乗算節点78がそれぞれ設けられている。第1の乗算節点77には、入力「z」および「x」からのそれぞれの入出力枝が接続されており、第2の乗算節点78には、入力「y」からの入出力枝および第1の乗算節点77からの入出力枝がそれぞれ接続されている。

【0062】このようなスケジューリング結果において、アロケーションされる加算器と減算器と比較器の遅延時間を、それぞれ5 (nsec)、乗算器の遅延時間を60 (nsec)、クロック周期を100 (nsec) とすると、第1のIF節点32において、条件が「true」の場合に実行される副CDFG34の遅延時間は5 (nsec)、条件が「false」の場合に実行される副CDFG35の遅延時間は65 (nsec) となり、条件の成立如何にかかわらず、1つのクロックステップで終了する。これに対して、第2のIF節

点72は、条件が「true」の場合に実行される副CDFG74の遅延時間は5 (nsec) であり、1クロックステップで終了するが、条件「false」の場合に実行される副CDFG75の遅延時間は120 (nsec) となり、クロック周期を超えるために、2つのクロックステップが必要となる。このため、回路全体としては、条件によっては3つのクロックステップ、または4つのクロックステップが必要になる。

【0063】図19は、図18の第1のIF節点32と第2のIF節点72とを、予め論理合成し、それぞれの組合せ論理回路に対応する節点を、第1のCIF節点39と第2のCIF節点79とによって示したものである。この場合、第1および第2のCIF節点39および79の遅延時間は、第1および第2のIF節点32および72それぞれにおける副CDFG34および35と、副CDFG74および75との長い方の遅延時間となり、第1のCIF節点39の遅延時間は、65 (nsec)、第2のCIF節点79の遅延時間は、120 (nsec) となる。このため、第2CIF節点79は、2つのクロックステップに跨ることになる。よって、図19におけるスケジューリング結果では、全体として3クロックステップが必要になる。

【0064】しかしながら、この場合に、条件が成立するとき（「true」）と、条件が成立しないとき（「false」）とにおいて、同一のクロックステップで終了するIF節点のみを予め論理合成すると、図19に示すCDFGは、第1のIF節点32のみが、CIF節点39に変換されて、図20に示すCDFGとなる。図20に示すCDFGでは、第2のIF節点72において、条件が成立する場合と条件が成立しない場合とによって、回路全体のクロック数は、それぞれ2つのクロックステップと3つのクロックステップとなる。図19に示すCDFGの場合には、条件に関わらず、3つのクロックステップを必要とするために、図20に示すCDFGでは、演算処理をより高速化することができる。

【0065】＜実施の形態3＞次に、本発明の高位合成方法の実施の形態のさらに他の例を、図21の動作記述に基づいて説明する。図21に示す動作記述において、関数「send(x)」は、変数xを通信路へ送信し、通信が完了するまで待機する動作を表す。すなわち、図21の動作記述は、条件 (if) 「 $d > e$ 」の場合、「 $x = a - b$ 」を通信路に転送し、その他の条件の場合 (else)、「 $x = a * (b - c)$ 」を通信路に転送することを表す。

【0066】図21の動作記述に対するCDFGの一例を図22に示す。クロックステップstep1には、比較を示す比較節点81がスケジューリングされており、クロックステップstep2には、IF節点82がスケジューリングされている。比較を示す比較節点81には、入力「d」および「e」からの入出力枝がそれぞれ

接続されている。比較節点81からの入出力枝によって、コントロール信号k1がIF節点82に与えられている。

【0067】IF節点82の内部には、第1および第2の副CDFG84および85が設けられている。第1の副CDFG84は、コントロール信号k1が「1」（真）の場合に実行され（「true」で表す）、第2の副CDFG85は、コントロール信号k1が「0」（偽）の場合に実行される（「false」で表す）。

【0068】「true」で表された副CDFG84は、入力「a」および「b」からの入出力枝が接続される減算節点86が設けられている。

【0069】「false」で表される副CDFG85には、入力「b」および「c」からの入出力枝がそれぞれ接続された第2の減算節点87と、第2の減算節点87からの入出力枝が接続される乗算節点88がそれぞれ設けられている。乗算節点87には、入力「a」からの入出力枝も接続されている。

【0070】クロックステップ3以降のクロスステップには、動作記述中の「send(x)」に対応する第1および第2のsend節点91および92がスケジューリングされている。第1のsend節点91には、減算節点86からの入出力枝が接続されており、第2のsend節点92には、乗算節点88からの入出力枝が接続されている。各send節点91および92には、それぞれ、1つのコントロール入出力枝が接続されている。

【0071】このように、通信を行うために、コントローラの制御が必要な第1および第2のsend節点91および92を、IF節点82の外部に取り出すことにより、IF節点82を、組合せ論理回路で実現することができる。従って、前記実施の形態1および2で説明したように、IF節点82を、予め論理合成しておくことができる。

【0072】send節点には、通常、1つのデータ入出力枝と、1つのコントロール入出力枝がそれぞれ接続されている。データ入出力枝には、通信路に転送するデータが入力される。コントロール入出力枝からは、データ転送が終了したことを示すコントロール信号が出力される。このために、図21に示される動作記述は、図23に示すようにスケジューリングされて、第1および第2のsend節点91および92は、IF節点82内における第1および第2の副CDFG84および85にそれぞれ配置される。各send節点91および92は、データ転送が終了するまで待機状態になるために、クロックステップstep3以降の複数のクロックステップにわたってデータの転送を実行する。

【0073】このように、各send節点91および92の動作が、コントローラによるコントロール信号によって制御されるために、send節点91および92を含むCDFG82は、コントローラとの間にてコントロール

信号を送受する必要がある。組合せ論理回路のみでは構成することができず、send節点91および92を有するIF節点82を、予め論理合成することはできない。

【0074】しかしながら、図22に示すように、各send節点91および92をIF節点82の外部に取り出すことにより、IF節点82を、組合せ論理回路で実現することができ、IF節点82を、予め論理合成しておくことが可能になる。

【0075】なお、このように、各send節点91および92をIF節点82の外部に取り出す場合には、図24に示すように、IF節点82の外部に取り出された各send節点91および92が同一機能であれば、1つのsend節点93のみを設けるようにしてもよい。

【0076】図25は、図24に示すIF節点82を、図13に示すように論理合成して組合せ回路とし、図15と同様に、対応するCIF節点94に入れ換えた場合のスケジューリング結果である。図25に示すスケジューリング結果では、図23に示すスケジューリング結果よりも、クロックステップ数が1つ少なくなり、処理速度をより高速化することができる。

【0077】なお、組合せ論理回路で表すことができない節点が、IF節点における副CDFGの最上位にある場合には、その節点を、IF節点の上側に取り出すようにしてもよい。

【0078】＜実施の形態4＞本発明の高位合成方法の実施の形態のさらに他の例を、図26の動作記述に基づいて説明する。図26に示す動作記述では、条件(i f)「d>e」の場合は、 $x = (a + b - c) * c + a$ であり、その他の条件の場合(e l s e)は、 $x = (b + c) * c + (b + c) + a$ である。

【0079】この動作記述をCDFG変換してスケジューリングした一例を図27に示す。図27に示すように、クロックステップstep1には、比較を示す比較節点101がスケジューリングされており、クロックステップstep2には、IF節点102がスケジューリングされている。比較節点101には、入力「d」および「e」からの入出力枝が接続されるとともに、コントロール信号kを与えるコントロール入出力枝173が接続されている。

【0080】また、クロックステップstep2にスケジューリングされたIF節点102は、内部に第1および第2の副CDFG104および105が設けられている。第1の副CDFG104は、IF節点を含むことができる。第1の副CDFG104は、コントロール信号kが「1」（真）の場合に実行され（「true」で表す）、第2の副CDFG105は、コントロール信号kが「0」（偽）の場合に実行される（「false」で表す）。

【0081】「true」で表された副CDFG104

には、入力「a」および「b」からの入出力枝が接続された第1の加算節点106が設けられており、第1の加算節点106の入出力枝は、第1の減算節点107に接続されている。第1の減算節点107には、入力「c」からの入出力枝が接続されている。第1の減算節点107の入出力枝は、第1の乗算節点108に接続されており、第1の乗算節点108に、入力「c」からの入出力枝が接続されている。第1の乗算節点108の入出力枝は、第2の加算節点109に接続されており、第2の加算節点109に、入力「a」からの入出力枝が接続されている。そして、第2の加算節点109からの入出力枝がIF節点102の入出力枝に接続されている。

【0082】「false」で表される副CDFG105には、第3の加算節点111が設けられており、第3の加算節点111には、入力「b」および「c」からの入出力枝が接続されている。第3の加算節点111からの入出力枝は、第2の乗算節点112および第2の減算節点113に接続されている。第2の乗算節点112には、入力「c」からの入出力枝が接続されており、第2の乗算節点112からの入出力枝は、第2の減算節点113に接続されている。第2の減算節点113からの入出力枝は、第4の加算節点114に接続されている。そして、第4の加算節点114からの入出力枝がIF節点102の入出力枝に接続されている。

【0083】図27に示すCDFGも、実施の形態1および2と同様に、条件が成立する場合（「true」）と、条件が成立しない場合（「false」）とにおいて、同一のクロックステップにおいて終了するIF節点のみを予め論理合成することにより、図28に示す回路とされる。この回路は、比較節点101に対して比較器121が割り当てられており、また、IF節点102組合せ回路に対して、CIF節点120が割り当てられている。

【0084】しかしながら、図27に示す回路構成では、IF節点の条件が成立した場合に実行される副CDFG104と、条件が成立しなかった場合に実行される副CDFG105の両方に、乗算節点108および112を含む場合には、図28に示すように論理合成すると、2つの乗算器122および123を含む組合せ論理回路が生成される。通常、乗算器は、加算器および減算器に比べて面積が大きいので、生成される回路全体の面積が増加することになる。

【0085】このような回路規模の増大を防止するためには、次の方法が実施される。この方法では、IF節点を、指定された節点の上側および下側において分割し、指定された節点をIF節点の外部に配置するようになっている。これにより、IF節点を予め論理合成した場合に、回路全体の面積が増加することが防止される。また、コントローラによる制御が必要なために組合せ論理回路を構成できない演算の節点を指定してIF節点の外

部に配置することにより、コントローラによる制御が必要な節点を含むIF節点についても、予め論理合成して組合せ論理回路にすることができる。

【0086】この方法を、図29に示すフローチャートに基づいて説明する。まず、通常の方法によって、動作記述をCDFGに変換し（図29のステップS21参照、以下同様）、変換されたCDFGに含まれる節点を、特定された節点の上側と下側において、それぞれグループに分割する（ステップS22）。

【0087】節点をグループ化する方法について、図30に示すCDFGに基づいて説明する。図30に示すCDFGは、2つの入力枝が接続された第1の加算節点131からの出力枝が、第1の減算節点132に接続されている。第1の減算節点132には、さらに、1つの入力枝が接続されており、出力枝が乗算節点133に接続されている。乗算節点133からの2つの出力枝が、それぞれ、第2の加算節点135および第3の加算節点136に接続されている。第1の加算節点131からの他の出力枝は、第2の減算節点134に接続されており、第2の減算節点134からの出力枝が第2の加算節点135に接続されている。

【0088】このようなCDFGにおいて、回路面積が大きい乗算節点133を、IF節点の外に出すことによって、回路規模が増大することを防止できる。この場合、まず、図30に示すように、CDFGに含まれる節点をグループ化する。節点をグループ化するには、IF節点の外に出す乗算節点133を指定して、その上側および下側において、各節点をグループ化する。乗算節点133の上側のグループAは、それぞれの出力が乗算節点133に直接的または間接的に入力される節点の集合であり、第1加算節点131および第1減算節点132が含まれている。第1加算節点131のように、その出力が、第1減算節点132を経て、乗算節点133の入力になるものも含む。

【0089】乗算節点133よりも下側のグループCに含まれる節点は、乗算節点133の出力を入力とする節点の集合であり、第3加算節点135および第4加算節点136が含まれる。この場合も、乗算節点133の出力が他の節点を経て入力される節点があれば、その節点も含まれる。グループBは、グループAにもグループCにも含まれない節点の集合であり、乗算節点133の演算と無関係な節点である。グループBには、第2減算節点134が含まれる。このようにして、IF節点における節点がグループ化される。

【0090】このようにして、節点がグループ化されると、各グループに基づいて、IF節点を分割し、指定された節点をIF節点の外部に配置する（図29のステップS23）。その後、外部に配置された節点の共通化と、不要な入出力枝を削除することにより、CDFGを単純化する（ステップS24）。その後は、前述した高

位合成方法と同様に、変更されたCDFGをスケジューリングして、アロケーションすることにより、回路を生成する。

【0091】このような高位合成方法を、図27のCDFGに基づいて説明する。図27に示すCDFGにおいても、前述したように、まず、IF節点102に含まれる節点をグループ化する。この場合、IF節点102における「true」で表される副CDFG104、および、「false」で表される副CDFG105のそれぞれに関して、回路規模の大きな第1の乗算節点108および第2の乗算節点112を中心として、それぞれ、節点のグループ化が実施される。

【0092】従って、第1の乗算節点108に関して、それよりも上側のグループAには、第1乗算節点108にそれぞれの出力が入力される第1加算節点106および第1減算節点107が含まれ、第1乗算節点108よりも下側のグループCには、第1乗算節点108からの出力が入力される第2加算節点109が含まれる。

【0093】同様に、第2の乗算節点112に関して、それよりも上側のグループAには、第2乗算節点112に出力が入力される第3加算節点111が含まれ、第2乗算節点112よりも下側のグループCには、第2乗算節点112からの出力がそれぞれ入力される第2減算節点113および第4減算節点114が含まれる。

【0094】このように、IF節点102内に含まれる節点がグループ化されると、図31に示すように、IF節点102を、グループAに含まれる各節点にて構成される第1のIF節点（図に「IF1」にて示す）141と、グループCに含まれる各節点にて構成される第2のIF節点（図に「IF2」にて示す）142の2つに分割する。

【0095】この場合、第1および第2の乗算節点108および112とは無関係の節点のグループBがあれば、グループAに含まれる節点数がグループCに含まれる節点数よりも少ない場合は、グループBに含まれる節点が、グループAにて構成される第1のIF節点141に含められ、そうでない場合は、第2のIF節点142に含められる。第1および第2の乗算節点108および112は、グループA、B、Cのいずれにも含まれないため、第1のIF節点141と第2のIF節点142の間に配置されることになり、図31に示すように、第1IF節点141および第2IF節点142の外部に配置される。

【0096】このようなCDFGが得られると、得られたCDFGが簡単化される。CDFGの簡単化は、前述したように、IF節点の外部に配置された複数の節点の演算が同一な場合に、それらの節点を1つの節点にて共通化すること、および、不要な入出力枝の削除で実施される。

【0097】図31に示すCDFGが得られると、第1

および第2のIF節点141および142の外部に配置された第1および第2の乗算節点108および112は、図32に示すように、1つの乗算節点143によって共通化する。これにより、2つの乗算節点108および112に対して、1つの乗算器が割り当てられることになる。この時、当然に、共通化される2つの節点は同一の演算器で行うことができる必要がある。2つの演算が異なる種類の場合には、節点同士の共通化は行われない。

【0098】次に、不要な入出力枝を削除する。不要な入出力枝は、各IF節点141および142内を、いずれの節点にも接続されることなく、すなわち演算に関係することなく通過する入出力枝であり、図32においては、第1IF回路141内の入出力枝144～147である。これらの入出力枝144～147が削除されることにより、図33に示すように、簡単化されたCDFGが得られる。

【0099】このようにして、分割された第1のIF節点141および第2のIF節点142が得られると、第1のIF節点141および第2のIF節点142を、それぞれ、前述したようにして論理合成して、図34

(a)に示すように、第2IF節点141に対応する第1のCIF節点151と、図34(b)に示すように、第2のIF節点142に対応する第2のCIF節点152とを、それぞれ生成する。

【0100】この場合、第1のCIF節点151には、第1および第2の加算器153および154と、第1の減算器155と、第1のセレクト156とが設けられている。第1の加算器153には、入力「a」および「b」が入力されており、第2の加算器154には、入力「b」および「c」が入力されている。第1の減算器155には、第1の加算器153の出力と入力「c」とが入力されている。そして、セレクト156には、第1の減算器155の出力と第2の加算器154の出力が入力されており、コントロール信号kによって、第1の減算器155の出力、第2の加算器154の出力のいずれかが出力される。従って、セレクト153からは、演算「 $a+b-c$ 」または「 $b+c$ 」のいずれかの結果が出力される。

【0101】第2のCIF152には、第3および第4の加算器157および158と、第2の減算器159と、第2のセレクト160とが設けられている。第3の加算器153には、入力「a」と、セレクト153からの出力である「 $(a+b-c) * c$ 」または「 $(b+c) * c$ 」のいずれかが入力されており、第2の減算器159には、乗算器143からの出力である「 $(a+b-c) * c$ 」または「 $(b+c) * c$ 」のいずれかと、第1CIF節点151の出力「 $b+c$ 」とが入力されている。第2の加算器158には、入力「a」と、第2減算器の出力とが入力されている。第2のセレクト1

60には、第3の加算器157の出力と第4の加算器154の出力が入力されており、コントロール信号kによって、第3の加算器157の出力、第4の加算器158の出力のいずれかが「x」として出力される。

【0102】各CIF151および152において、それぞれの加算器と減算器の遅延時間をそれぞれ5 (nsec)、乗算器の遅延時間を60 (nsec) とすると、第1CIF151の遅延時間は10 (nsec)、第2CIF152の遅延時間も10 (nsec) となる。

【0103】第1のCIF151および第2のCIF152を用いたCDFGを、図35に示す。このCDFGの最大遅延パスは、比較節点101、第1のCIF151、乗算節点143、第2のCIF152を通るパスであり、クロック周期が100 (nsec)、比較器の遅延時間が5 (nsec) であるとすると、最大遅延時間は85 (nsec) となり、クロック周期を超えることがなく、1クロックにおいて実行することが可能である。

【0104】図27に示すCDFGは、IF節点も含めると10個の節点より構成されているのに対して、図35に示すCDFGは4つの節点しか含まれず、高位合成を高速で行うことが可能である。

【0105】また、図27に示すCDFGの実行には、2クロック周期が必要であるのに対して、図35に示すCDFGは、1クロック周期で実行することができ、高速処理が可能になる。最終的には、図35に示すCDFGから、図36に示す回路を得ることができる。図27に示すCDFGから生成された図28に示す回路では、2つの乗算器122を含むのに対して、図27に示すCDFGから生成された図36に示す回路では、1つ乗算器161しか含まず、回路の面積が小さくなっている。

【0106】<実施の形態5>次に、本発明の高位合成方法の実施形態の他の例について説明する。まず、図37に示す動作記述を前述した実施の形態1および2と同様にして高位合成した場合について説明する。図37に示す動作記述では、条件 (if) 「d>e」の場合は、 $x = (a+b) * b + a$ であり、その他の条件の場合 (else) は、 $x = (b+c) * a - (a+c)$ である。

【0107】この動作記述をCDFG変換してスケジューリングした一例を図38に示す。図38に示すように、クロックステップstep1には、比較を示す比較節点171がスケジューリングされており、クロックステップstep2には、IF節点172がスケジューリングされている。比較節点171には、入力「d」および「e」からの出力枝が接続されるとともに、コントロール信号kを与えるコントロール出力枝173が接続されている。

【0108】また、クロックステップstep2にスケ

ジューリングされたIF節点172は、内部に第1および第2の副CDFG174および175が設けられている。第1の副CDFG174は、コントロール信号kが「1」(真)の場合に実行され(「true」で表す)、第2の副CDFG175は、コントロール信号kが「0」(偽)の場合に実行される(「false」で表す)。

【0109】「true」で表された副CDFG174には、入力「a」および「b」からの入出力枝が接続された第1の減算節点176が設けられており、第1の加算節点176からの出力枝は、第1の乗算節点177に接続されている。第1の乗算節点177には、入力「b」からの出力枝が接続されている。第1の乗算節点177の入出力枝は、第1の加算節点178に接続されており、第1の加算節点178に、入力「a」からの出力枝が接続されている。第1の加算節点178からの出力枝がIF節点172の出力枝に接続されている。

【0110】「false」で表される副CDFG175には、第2の加算節点181が設けられており、第2の加算節点181には、入力「b」および「c」からの出力枝が接続されている。第2の加算節点181からの出力枝は、第2の乗算節点182および第2の減算節点183に接続されている。第2の乗算節点182には、入力「a」からの出力枝が接続されており、第2の乗算節点182からの出力枝は、第2の減算節点183に接続されている。第2の減算節点183からの出力枝がIF節点172の出力枝に接続されている。

【0111】このようなCDFGのIF節点172は、前述したようにして論理合成することにより、図39に示すように、CIF184とされる。このようなCDFGによって、図40に示す回路が得られる。この回路は、入力「a」および「b」が入力される第1減算器191を有しており、第1減算器191の出力が第1乗算器192に入力されている。第1乗算器192には、入力「b」も入力されており、第1乗算器192の出力が第1加算器193に入力されている。第1加算器193には、入力「a」も入力されている。第1加算器193の出力は、セクタ197に与えられている。

【0112】また、この回路には、第2加算器194入力「a」および「b」が入力される第2加算器194が設けられており、第2加算器194の出力が第2乗算器195に入力されている。第1乗算器195の出力は、第2減算器196に入力されている。第2減算器196には、第2加算器194の出力が入力されており、第2減算器196の出力がセクタ197に与えられている。セクタ197は、比較器198からのコントロール信号kに基づいて、第1加算器193の出力、第2減算器196の出力のいずれかを出力する。

【0113】また、このような回路においては、次の(2)式を満足する演算器があれば、その演算器に關す

る節点をIF節点の外部に配置することによって、回路面積をさらに低減することができる。

$$(\text{演算器の入力数}) \times (\text{セレクトアの面積}) \leq (\text{演算器の面積}) \cdots (2)$$

すなわち、IF節点に含まれる副CDFGを構成する節点に対応する演算器において、同一の演算器が複数存在する場合には、その演算器の1つの面積が、セレクトアの面積に対して、演算器の入力数倍以上であれば、その演算器に対応する節点をIF節点の外部に配置する。

【0115】例えば、図40に示す回路において、乗算器192および195に対して、「true」と「false」の2通りの入力を与えるために、各乗算器192および195の入力の個数分(2個分)のセレクトアが必要になるが、このような個数のセレクトアを増加することによって、一方の乗算器を削除することができるために、必要とされるセレクトアの全面積に対して、削除される乗算器の面積が大きければ、乗算器を削除することによって、セレクトアが増加するにもかかわらず、全体の回路面積を低減させることができる。

【0116】図40に示す回路において、乗算器の面積が10、加算器および減算器の面積が2、セレクトアの面積が2の割合にそれぞれなっているものとする、図38に示すIF節点172の副CDFG174および175に含まれる各節点に対応するそれぞれの演算器において、前記(2)式を満たすものは、図40に示す乗算器192および195である。

【0117】このために、各乗算器192および195に対応する乗算節点177および182を、実施の形態3で説明したようにして、IF節点172の外部に配置する。

【0118】図41は、図38に示すIF節点172を分割したCDFGである。このCDFGは、第1および第2の乗算節点177および182を指定して、各乗算節点177および182の上側の節点のグループと、下側の節点のグループとに分割することによってIF節点172を分割して、第1IF節点201および第2IF節点202とする。これにより、各乗算節点177および182が、第1IF節点201および第2IF節点202の外部に配置される。そして、外部に配置された各乗算節点177および182を1つの乗算節点203によって共通化し、不要な入出力枝を削除することにより、図41に示すCDFGが得られる。

【0119】このように、必要とされるセレクトアの面積に対して、削除された乗算器の面積が大きいことにより、回路面積を低減させることができる。

【0120】図42は、図41に示すCDFGにおける第1のIF節点201と第2のIF節点202とを、前述したように、組合せ論理回路である第1CIF203と、第2CIFとに置き換えたCDFGのスケジューリング結果を示す。また、図43は、図42に示すスケジューリング結果を回路化したものである。

【0114】

【0121】図43に示す回路では、第1のCIF204に対応する第1の組合せ論理回路221および第2のCIFに対応する第2の組合せ論理回路222と、両組合せ論理回路221および222の間に配置された乗算器212とが設けられている。第1の組合せ論理回路221には、入力「a」および「b」がそれぞれ入力される第1の減算器211および第1の加算器214が設けられており、第1減算器211および第1の加算器214の出力が、それぞれ、第1のセレクトア217に入力されるとともに、第2のセレクトア218に入力されている。

【0122】第1および第2のセレクトア217および218は、比較器223からのコントロール信号kによって、それぞれ、第1減算器211および第1の加算器214の出力のいずれかを選択して出力する。第1のセレクトア217および第2のセレクトア218の出力は、それぞれ乗算器212に与えられている。

【0123】乗算器212の出力は、第2の組合せ論理回路222に設けられた第2の加算器213および第2の減算器216にそれぞれ与えられており、第2の加算器213および第2の減算器216の出力が第3のセレクトア219に与えられている。第3のセレクトア219は、比較器223からのコントロール信号kによって、それぞれ、第2の加算器213および第2の減算器216の出力のいずれかを選択して出力する。そして、第3セレクトア219の出力が、演算結果の出力「x」になる。

【0124】図43に示す回路を、図40に示す回路と比較すると、図40に示す回路は、2つの乗算器192および195が設けられているが、1つのセレクトア197しか設けられていない。これに対して、図43に示す回路では、1つの乗算器212しか設けられていないが3つのセレクトア217~219が設けられている。この場合、1つの乗算器の面積は、増加するセレクトアの面積よりも大きく、前記(2)式を満たしているため、回路全体では、図43に示す回路の面積が、図40に示す回路の面積よりも小さくなる。

【0125】(2)式を満たす節点が「true」または「false」の副CDFGに複数ある場合は、その節点の演算を実行する演算器の面積が最も大きい演算をIF節点が外部に配置されるように、上記方法を適用する。そして、分割して得られる2つのIF節点について、再度上記方法を適用する。

【0126】IF節点の外部に配置された節点(図42の場合は乗算節点203)と別の節点が、同一の演算器を共通化するような場合には、図40に示す回路の面積がさらに小さくなる。この場合、演算器の面積が(2)

式を満たす程度に大きくなっていなくても、面積削減の効果が現れる。このような可能性を考慮して、条件を、次の(3)式のようにしてもよい。ここで、定数は1よ

$$(\text{演算器の入力数}) \times (\text{セクタの面積}) \times (\text{定数}) \leq (\text{演算器の面積})$$

... (3)

【0128】

【発明の効果】本発明の請求項1記載の高位合成方法では、CDFGの一部である複数の節点が、1つの節点として扱われるため、CDFGが含む節点が少なくなり、高位合成に必要な処理時間が短くなる。また、節点のCDFGの一部を予め論理合成して最適化しているため、置き換えられた1つの節点の遅延時間には、最終的な回路における遅延時間を用いることができ、遅延時間の見積り精度が良くなる。

【0129】本発明の請求項2記載の高位合成方法では、条件分岐に対応するCDFGの部分は、条件分岐のコントロール信号も含めて予め論理合成されるため、高位合成に必要な処理時間が短くなり、遅延の見積り精度が向上する。また、合成した回路が含むセクタが減少することによって、回路規模を小さくすることができる。さらに、コントローラによって条件分岐の制御を行う信号を生成する必要がなくなるために、回路規模を小さくすることができる。また、条件分岐の条件判定と、条件分岐自体の動作を同一クロックステップで実行できるように、高位合成によって得られる回路の動作を高速化することもできる。

【0130】本発明の請求項3記載の高位合成方法では、分岐条件が真でも偽でも無駄なクロックステップが発生するおそれがなく、高位合成によって得られる回路の動作を高速化することができる。

【0131】本発明の請求項4記載の高位合成方法では、CDFGがコントローラとの信号の送受に関する演算を含むため、予め論理合成して組合せ論理回路にできない場合、あるいは、CDFGが大規模な演算器を含むために、予め論理合成して組合せ論理回路とすると回路規模が不必要に大きくなるような場合においても、回路規模の縮小および生成される回路動作の高速化が可能になる。

【0132】本発明の請求項5記載の高位合成方法も、CDFGがコントローラとの信号の送受に関する演算を含むため、予め論理合成して組合せ論理回路にできない場合、あるいは、CDFGが大規模な演算器を含むために、予め論理合成して組合せ論理回路とすると回路規模が不必要に大きくなるような場合においても、回路規模の縮小および生成される回路動作の高速化が可能になる。

【0133】本発明の請求項6記載の高位合成方法では、面積が大きい演算器を含む動作記述についても、回路規模の縮小および生成される回路動作の高速化が可能になる。

り小さい数値で、回路によって実験的に求められる値である。

【0127】

【0134】本発明の請求項7記載の記録媒体では、このような高位合成方法を容易に実行することができる。

【図面の簡単な説明】

【図1】高位合成の手順を示すフローチャートである。

【図2】(1)式の動作記述のCDFGの一例を示す模式図である。

【図3】図2のCDFGのスケジューリング結果の一例を示す模式図である。

【図4】図3に示すスケジューリング結果に対応した回路の構成を示す模式図である。

【図5】(1)式の動作記述のCDFGをスケジューリングした結果の他の例を示す模式図である。

【図6】図5に示すスケジューリング結果に対応した回路の構成を示す模式図である。

【図7】動作記述の他の例である。

【図8】図7の動作記述のCDFGをスケジューリングした結果を示す模式図である。

【図9】図8に示すスケジューリング結果に対応した回路の構成を示す模式図である。

【図10】高位合成における遅延時間の説明図である。

【図11】(a)および(b)は、それぞれ、論理合成において最適化した回路を示す模式図である。

【図12】本発明の高位合成方法の実施の形態の一例を示すフローチャートである。

【図13】その高位合成方法において、CDFGに含まれるIF節点を、論理合成して生成された組合せ論理回路を示す模式図である。

【図14】図13の組合せ論理回路を示す節点である。

【図15】図14に示す節点を用いたCDFGである。

【図16】図15に示すCDFGに対応した回路を示す模式図である。

【図17】動作記述の一例を示す式である。

【図18】図17に示す動作記述のCDFGの一例を示す模式図である。

【図19】そのCDFGをCIF節点に置き換えたCDFGの一例を示す模式図である。

【図20】そのCDFGのIF節点をCIF節点に置き換えたCDFGの他の例を示す模式図である。

【図21】動作記述の他の例を示す式である。

【図22】図21に示す動作記述のCDFGの一例を示す模式図である。

【図23】そのCDFGにおける節点をIF節点の外部に配置したCDFGの一例を示す模式図である。

【図24】図23のCDFGを最適化したCDFGを示す模式図である。

【図25】そのCDFGのIF節点をCIF節点に置き換えたCDFGの一例を示す模式図である。

【図26】動作記述の他の例を示す式である。

【図27】図26に示す動作記述のCDFGの一例を示す模式図である。

【図28】図27に示すCDFGに対応した回路を示す模式図である。

【図29】本発明の高位合成方法の実施の形態の他の例を示すフローチャートである。

【図30】節点をグループ分けする場合を説明するための模式図である。

【図31】図26に示すCDFGを分割したCDFGを示す模式図である。

【図32】図31に示すCDFGを最適化したCDFGを示す模式図である。

【図33】図32に示すCDFHを単純化したCDFGを示す模式図である。

【図34】(a) および (b) は、それぞれ、図33に示すCIF節点に対応した回路を示す模式図である。

【図35】そのCDFGをCIF節点に置き換えたCDFGの一例を示す模式図である。

【図36】図27に示すCDFGに対応した回路を示す模式図である。

【図37】動作記述の他の例を示す式である。

【図38】図37に示す動作記述のCDFGの一例を示す模式図である。

【図39】そのCDFGをCIF節点に置き換えたCDFGの一例を示す模式図である。

【図40】図39に示すCDFGに対応した回路を示す模式図である。

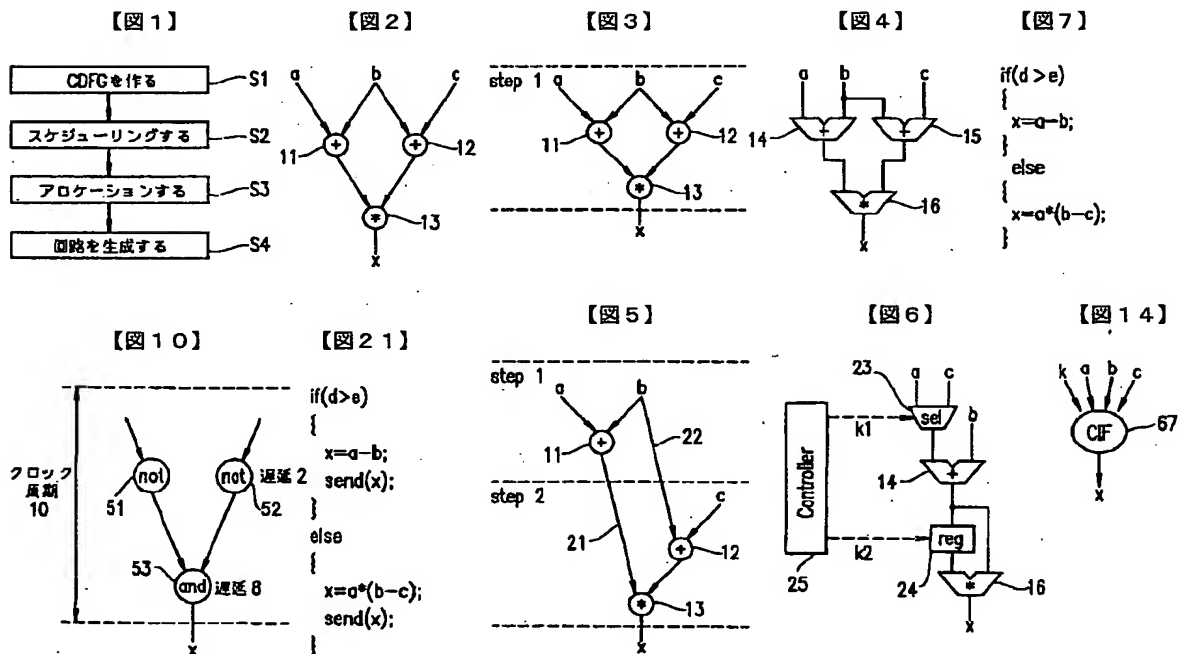
【図41】図38に示すCDFGを分割したCDFGの一例を示す模式図である。

【図42】そのCDFGをCIF節点に置き換えたCDFGの一例を示す模式図である。

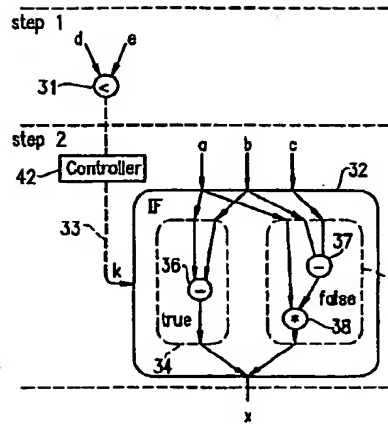
【図43】図39に示すCDFGに対応した回路を示す模式図である。

【符号の説明】

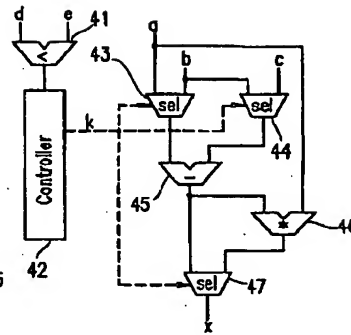
61、62 部分回路
63、64、159 減算器
65、161 乗算器
66、156 セレクタ
67、79、94、151、152、184 CIF節点
71、81、171、198 比較節点
72、82、102、141、142、201、202 IF節点
74、75、84、85、104、105、174、175 副CDFG
76、86、87、107、113、134、176、183、191、196 減算節点
77、78、88、108、112、132、133、177、182、192、195 乗算節点
91、92、93 send節点
106、109、111、114、131、135、136、153、154、157、158、178、181、193、194 加算器



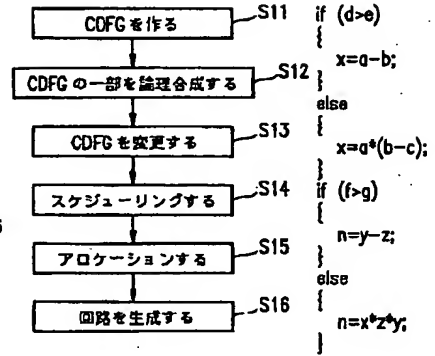
【図8】



【図9】

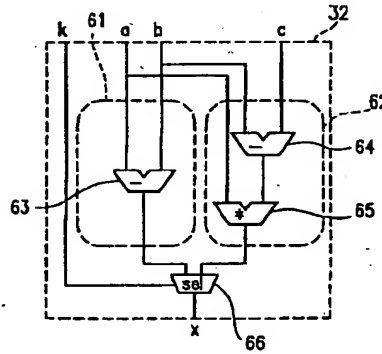


【図12】

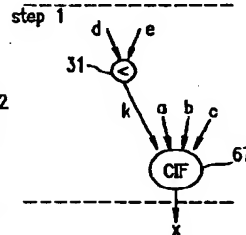


【図17】

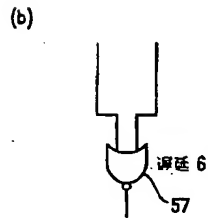
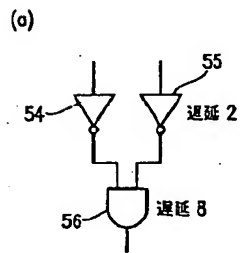
【図13】



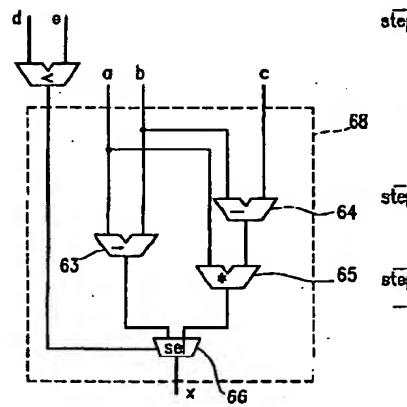
【図15】



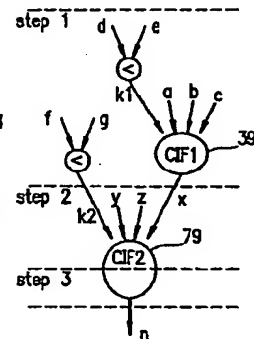
【図11】



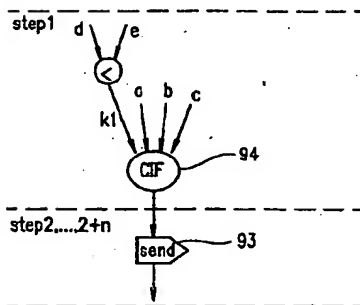
【図16】



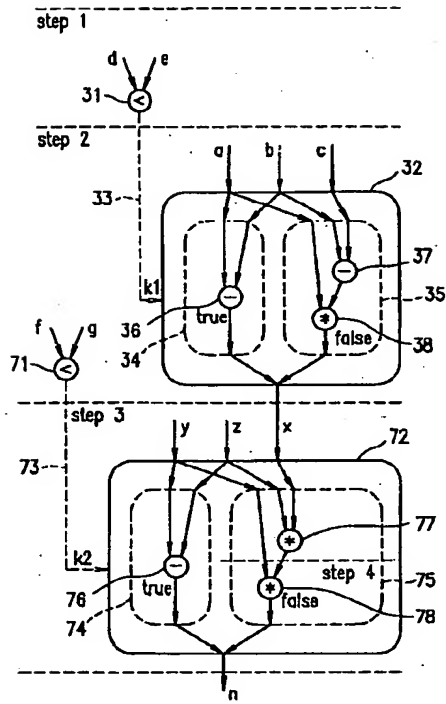
【図19】



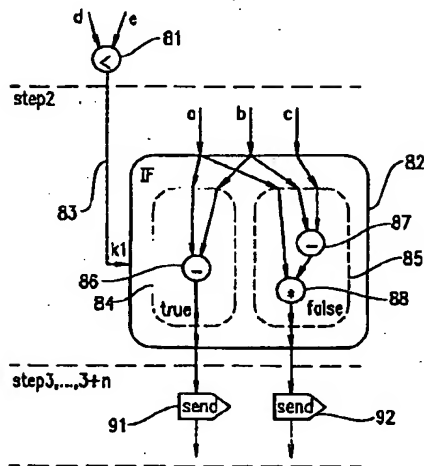
【図25】



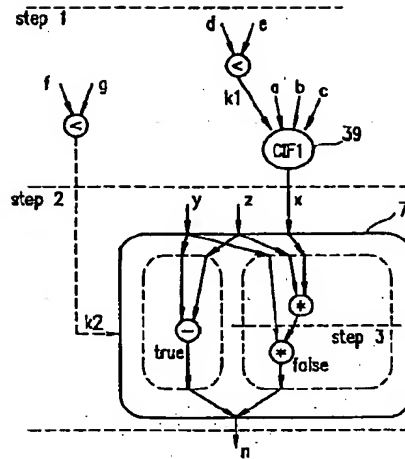
【図18】



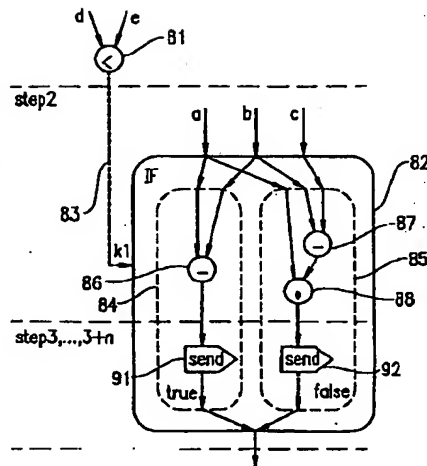
【図22】



【図20】



【図23】



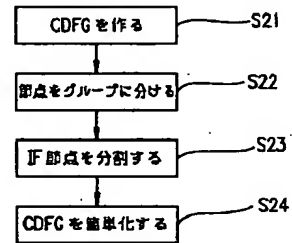
【図26】

```

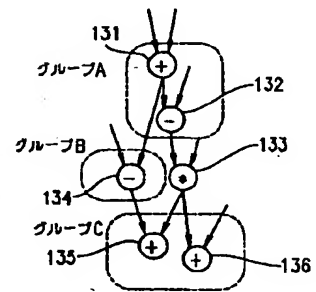
if(d>e)
{
  x=(a+b-c)*c+a;
}
else
{
  x=-(b+c)*c+(b+c)+a;
}

```

【図29】



【図30】



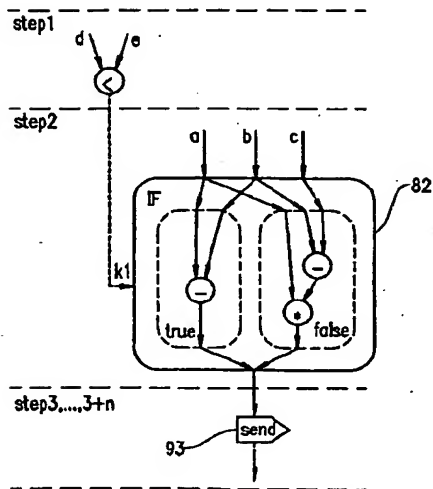
【図37】

```

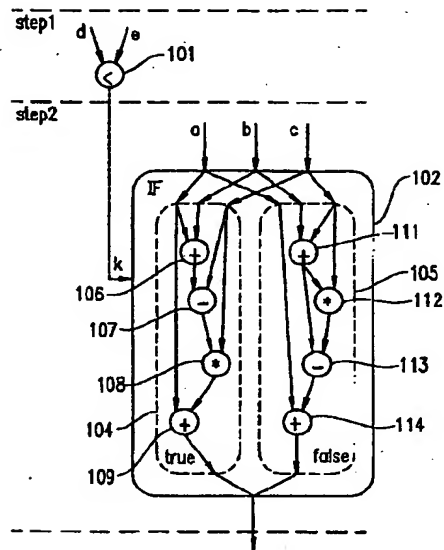
if (d>e)
{
  x=(a-b)*b+a;
}
else
{
  x=(a+b)*a-(a+b);
}

```

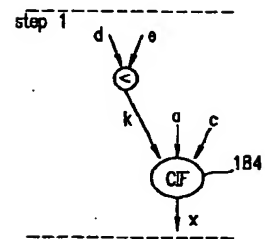
【図24】



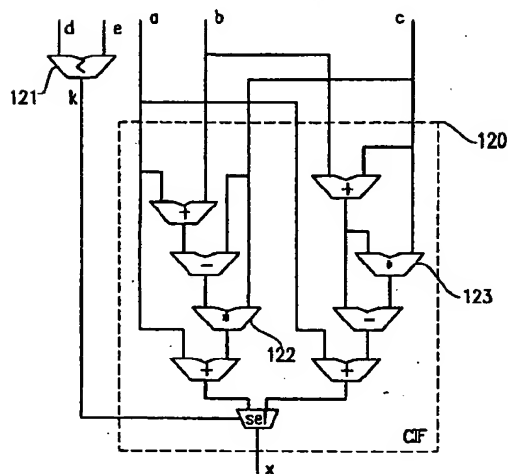
【図27】



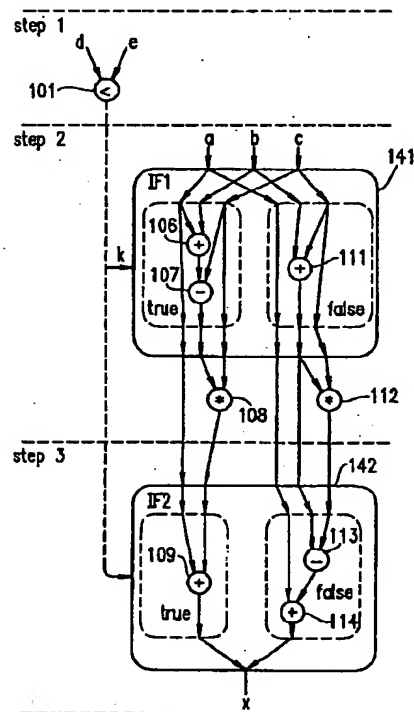
【図39】



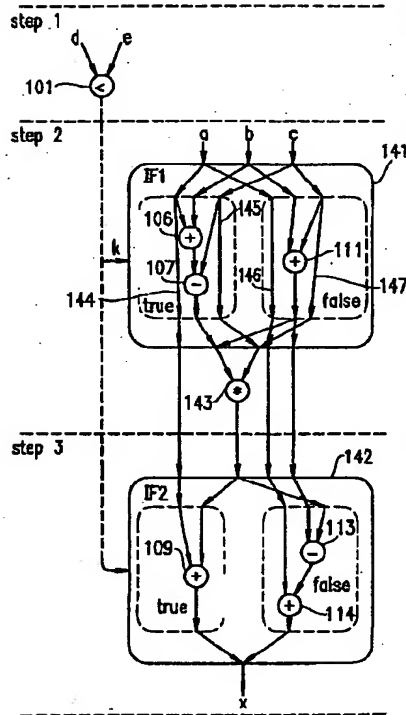
【図28】



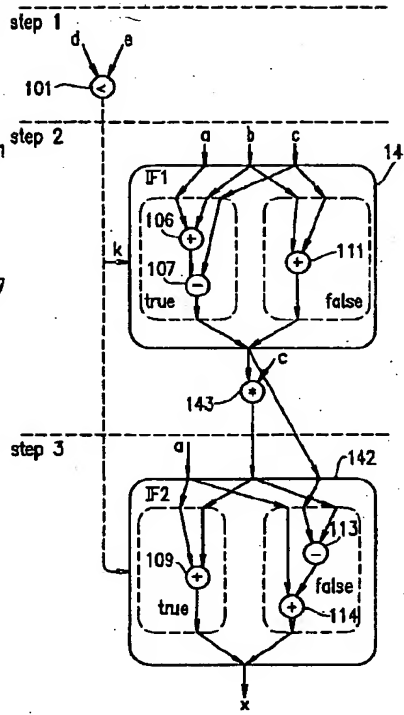
【図31】



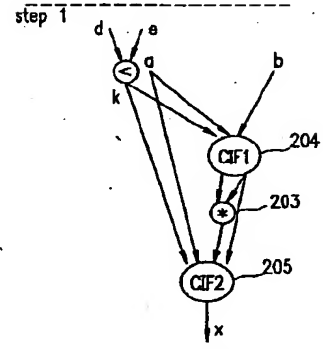
【図32】



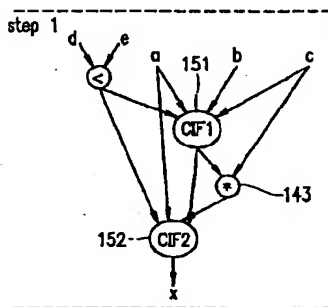
【図33】



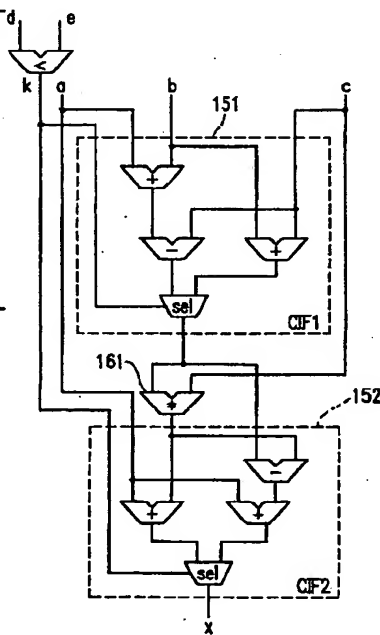
【図42】



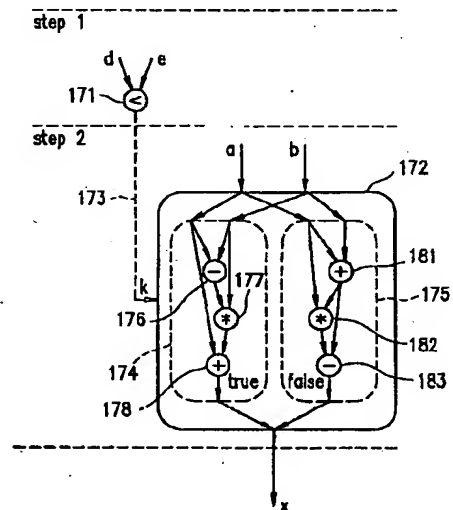
【図35】



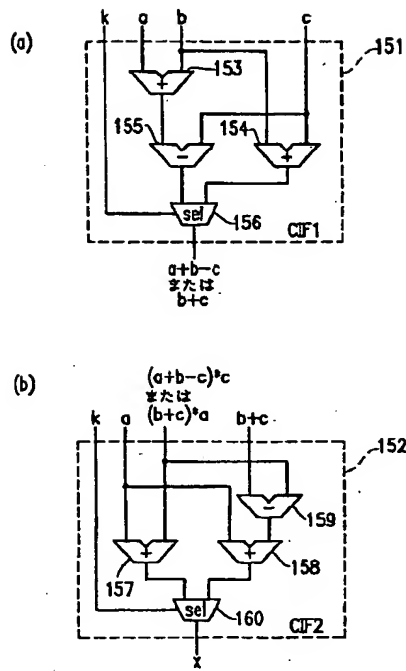
【図36】



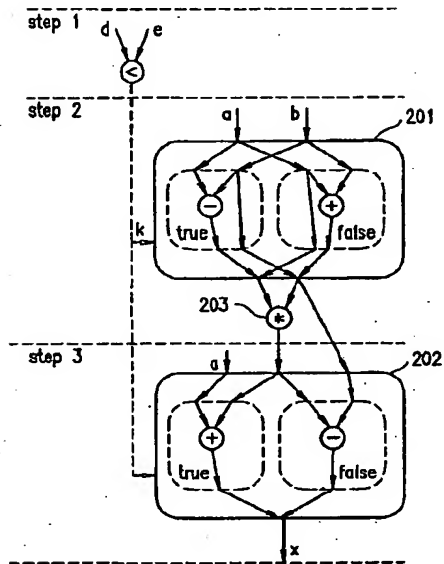
【図38】



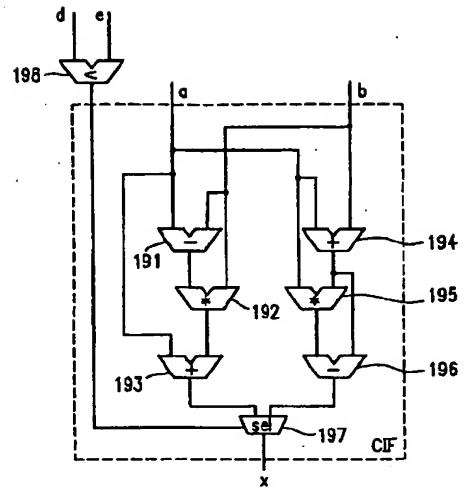
【図34】



【図41】



【図40】



【図43】

